UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/766,431	01/28/2004	David P. Vronay	MS306697.1/MSFTP543US	3594

27195 7590 06/05/2009

TUROCY & WATSON, LLP
127 Public Square
57th Floor, Key Tower
CLEVELAND, OH 44114

EXAMINER	
NGUYEN, PHILLIP H	
ART UNIT	**PAPER NUMBER**
2191	
NOTIFICATION DATE	**DELIVERY MODE**
06/05/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

Office Action Summary	Application No.	Applicant(s)	
	10/766,431	VRONAY ET AL.	
	Examiner	**Art Unit**	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _08 April 2009_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-3,5,7-9,11-22,24-27,29-31,35,36 and 39-45_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-3,5,7-9,11-22,24-27,29-31,35,36 and 39-45_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 4/8/2009.

2. Claims 1-3, 5, 7-9, 11-22, 24-27, 29-31, 35, 36, and 39-45 remain pending in this application with claims 1, 8, 27, and 29 amended, and claims 4, 6, 10, 23, 28, 32-34, 37, and 38 cancelled.

Response to Amendment

3. The rejection to claim 1 under 35 U.S.C. 112, second paragraph is hereby withdrawn in view of applicants' amendment to remove the deficiency.

Response to Arguments

4. Applicant's arguments with respect to claims 1-3, 5, 7-9, 11-22, 24-27, 29-31, 35, 36, and 39-45 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-3, 5, 7, 27, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Connelly et al. (U.S. Patent No. 7,376,733, hereinafter Connelly), in view of Eatough (U.S. Pub. No. 2002/0087623).

As per claims 1 and 27

Connelly teaches an interactive computer-implemented system for specifying and executing temporal order events (see at least col. 45-48 "**an embodiment of the present invention is directed to a method, system, computer program product, or apparatus that schedules and executes events within a network in real-time**"), comprising a processor executing:

a constraint component that receives loose temporal constraints associated with a plurality of events (see at least col. 5:27-28 "**sets the timing within each event object to the scheduled time of occurrence of the corresponding event**"), wherein the loose temporal constraints specify information about execution of an event comprising a start time or a stop time (see at least col. 7:8-12 "**Each event object 255 comprises at least three fields: an event resolution field, which defines when (daily, weekly, etc.) an event should occur periodically; an actually scheduled time of occurrence field, which contains the actual time when the next occurrence of an event is scheduled to occur…**") and event execution relative to other events (*Note: the date and time of each event also indicate which event executes first*); and

an order component that determines, a plurality of event execution orders in accordance with the loose temporal constraints (see at least col. 8:27-30 "**The event**

queue 260, which is part of the sorted queue 270, automatically sorts the event objects 255 in order by their next scheduled time or time and date of event occurrence"), selects an event execution order from the plurality of event execution orders (see at least col. 10:6-8 "**If there is an event due at step 620, then at step 630, the corresponding event object 255 is removed from the event queue 260**").

Connelly does not explicitly teach

a system information component that receives execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors; and

an order component that determines, a plurality of event execution orders via utility-based analysis of the execution system information, select an optimal event execution order from the plurality of event execution orders.

However, Eatough teaches

a system information component that receives execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors (see at least Para [0014] "**…transmission of a bandwidth** (*i.e. data throughput rate)* **PING upon receipt of an echo response from the network resource**"); and

an order component that determines, a plurality of event execution orders via utility-based analysis of the execution system information (see at least Para [0016] "**A**

priority value, in this embodiment, is assigned based at least in part of the relative amounts of bandwidth (i.e. *data throughput rate*) **to be employed in processing each task**”), select an optimal event execution order from the plurality of event execution orders (see at least Para [0031] "**processing said network related tasks based at least in part on the priority values, and the available network bandwidth**” – *Note: The tasks are ordered in accordance with the available bandwidth for efficient management purposes. Thus, the task with highest priority order is the optimal task order*).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Connelly’s approach to incorporate the teaching of Eatough for further scheduling the events based on the bandwidth (*i.e. throughput rate*) availability. The modification would have been obvious to an ordinary skill in the art because it would provide an efficient management method for processing each given task.

<u>As per claim 2:</u>

Connelly teaches

wherein the constraint is an event start and/or stop time (see at least col. 7:8-12 “**Each event object 255 comprises at least three fields: an event resolution field, which defines when (daily, weekly, etc.) an event should occur periodically; an**

actually scheduled time of occurrence field, which contains the actual time when the next occurrence of an event is scheduled to occur…").

As per claim 3:

Connelly teaches

wherein the constraint is event duration and/or a filter (see at least col. 3:5-16 "**…signifies that the "event" is to occur every 15 minutes**").

As per claim 5:

Eatough teaches

a system information component that provides information about an execution system to the order component to facilitate selection of an optimal event order (see at least Para [0014] "**…transmission of a bandwidth** (*i.e. data throughput rate)* **PING upon receipt of an echo response from the network resource**").

As per claim 7:

Eatough teaches

the information about an execution system includes data throughput rate (see at least Para [0014] "**…bandwidth** (*i.e. data throughput rate*)").

As per claim 43:

Eatough teaches

Wherein the execution system information comprises at least available memory, cache coherency, data throughput and number of processors of the system (see at least Para [0014] "**…bandwidth** (*i.e. data throughput rate*)").

7. Claims 8, 9, 12-22, 24, 26, 29-31, 35, 39, 44, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Colle et al. (U.S. Publication No. 2004/0133889, hereinafter Colle), in view of Eatough (U.S. Publication. No. 2002/0087623).

<u>As per claim 8</u>

Colle teaches an interactive computer-implemented system for specifying and executing temporal order events (see at Para [0025]), comprising a processor executing:

a display component that provides a plurality of object workspaces, the workspaces are user interfaces including a past, present and future space (see at least **FIGS. 3, 5-8, and 11**), the present space is an editable area (see at least Para [0064] "**In some implementations, the processor may display a user interface that allows a user to modify the estimated dates of the service items or the service planned start date or the service planned end date**"), the past and future space specify temporal constraints associated with a plurality of events (see least **FIGS. 3, 5-8, and 11**; see also at least Para [0054] "**The task list 310 includes a service order header 320 that identifies the service order by a title and the total duration** (*i.e. temporal constraints*) **required to perform the service (here, "581 hours")**"; and

a design component that temporally associates and disassociate objects in the editable area generating a plurality of event execution orders (see at least Para [0064] "…**allows a user to modify the estimated dates of the service items or the service planned start date or the service planned end date**" – *Note: the user modifies the planned start date and end date to generate execution orders for each task*) and determines an optimal execution order of events from the plurality of event execution orders based at least in part on the object associations specifying temporal constraints (see at least Para [0060] "**the processor determines an estimated start date and an estimated end date for each task in the service order based on the service planned start date, the duration of the task, and any task sequence dependencies identified between tasks**" – *Note: The tasks are ordered in an optimal execution order in order to effectively manage resource utilization and improve customer satisfaction of the service delivery*).

Colle does not explicitly teach

wherein non-associated objects order of execution is determined, via utility-based analysis of an executing system information comprising available memory, cache coherency, data throughput and number of processors.

However, Eatough teaches

wherein non-associated objects order of execution is determined, via utility-based analysis of an executing system information comprising available memory, cache

coherency, data throughput and number of processors (see at least Para [0016] "**A priority value, in this embodiment, is assigned based at least in part of the relative amounts of bandwidth** (i.e. *data throughput rate*) **to be employed in processing each task**").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Connelly's approach to incorporate the teaching of Eatough for further scheduling the events based on the bandwidth (*i.e. throughput rate*) availability. The modification would have been obvious to an ordinary skill in the art because it would provide an optimal method for processing each given task.

<u>As per claim 9:</u>

Colle teaches

object workspace that facilitate a graphical-based approach to specify relationships amongst objects (see at least **FIGS. 3, and 5-8**).

<u>As per claim 12:</u>

Colle teaches

the design component comprising a specification component that receives hard start and/or end times for events associated with objects (see at least Para [0027] "**The resource-independent scheduling function 135 for a service request takes into**

consideration a desired start date or a desired completion date for the service request”).

<u>As per claim 13:</u>

Colle teaches

the design component temporally associates objects as a function of respective location in the editable area (see at least **FIGS. 3, and 5-8** – *Note: Objects in the FIGS. are associated as functions to provide user with general knowledge about the ability of the resources to take on more work*).

<u>As per claim 14:</u>

Colle teaches

a duration component that receives information regarding event duration (see at least Para [0055] “**The task list 310 includes a service order header 320 that identifies the service order by a tile and the total duration**”).

<u>As per claim 15:</u>

Colle teaches

the design component receives and executes information related to nested events associated with respective objects (see at least Para [0054] “**…a graphical depiction 340 of the tasks and their interrelationships on the right side. The task list 310 displays a hierarchical ordering of tasks, illustrating the dependencies**

between tasks within the service order. The ability to organize tasks into a hierarchy of dependencies may be particularly useful for service orders that have a large number of tasks").

<u>As per claim 16:</u>

Colle teaches

a policy component that applies pre-defined rules to execution of the objects (see at least Para [0058] "**...automatically generate a date for a service request based on a desired start date for the service request, tasks to be performed for the service request, a predetermined duration for each task, and a sequence in which the task must be performed**').

<u>As per claim 17:</u>

Colle teaches

a policy component that applies predefined rules to editing of the objects (see at least Para [0064] "**...This may provide the user with general knowledge about the ability of the resources to take on more work. Using such general knowledge, the user may modify the estimated service schedule accordingly. This may help increase the accuracy of the resource-independent scheduling**").

<u>As per claim 18:</u>

Colle teaches

the design component receives and executes information regarding hierarchical relationship of respective objects (see at least Para [0054] "**…a graphical depiction 340 of the tasks and their interrelationships on the right side. The task list 310 displays a hierarchical ordering of tasks, illustrating the dependencies between tasks within the service order. The ability to organize tasks into a hierarchy of dependencies may be particularly useful for service orders that have a large number of tasks**").

<u>As per claim 19:</u>

Colle teaches

the design component receives and executes information regarding dependency relationship of respective objects (see at least Para [0054] "**…a graphical depiction 340 of the tasks and their interrelationships on the right side. The task list 310 displays a hierarchical ordering of tasks, illustrating the dependencies between tasks within the service order. The ability to organize tasks into a hierarchy of dependencies may be particularly useful for service orders that have a large number of tasks**").

<u>As per claim 20:</u>

Colle teaches

a query component that searches for events satisfy a query and displays objects associated with the event in temporal order (see at least **FIGS. 3 and 5-8**).

<u>As per claim 21:</u>

Colle teaches

the query component provides context information for respective objects (see at least **FIGS. 3 and 5-8**).

<u>As per claim 22:</u>

Colle teaches

objects placed in the past area are executed prior to objects in the present area and objects placed in the future area are executed after objects in the present area (see at least **FIG. 8** – *Note: tasks are executed in order as scheduled*).

<u>As per claims 24 and 35:</u>

Colle teaches

the design component associated objects in a non-linear conditional manner (see at least **FIGS. 3 and 5-8**).

<u>As per claim 26:</u>

Colle teaches

the design component associates objects based on a specified version (see at least **FIG. 8** - *Note: service order is considered task version*).

As per claim 29:

Colle teaches a method for object authoring implemented on a computer comprising the following computer executable instructions stored on a tangible computer readable medium:

receiving object data associated with events from a workspace including at least one of a past, present, or future area (see at least **FIGS. 3 and 5-8**);

associating objects temporally based at least in part upon relative object locations (see at least Para [0011] "**scheduling the performance of service actions that involve activities at multiple locations**"); and

generating a plurality of event execution orders based at least on the temporal association of the objects (see at least Para [0060] "**the processor determines an estimated start date and an estimated end date for each task in the service order based on the service planned start date, the duration of the task, and any task sequence dependencies identified between tasks**").

Colle does not explicitly teach

selecting an execution order of events from the plurality of event execution orders based at least on information regarding an execution system that executes the events, the information comprising available memory, cache coherency, data throughput and number of processors.

However, Eatough teaches

selecting an execution order of events from the plurality of event execution orders based at least on information regarding an execution system that executes the events, the information comprising available memory, cache coherency, data throughput and number of processors (see at least Para [0016] "**A priority value, in this embodiment, is assigned based at least in part of the relative amounts of bandwidth** (i.e. *data throughput rate*) **to be employed in processing each task**").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Connelly's approach to incorporate the teaching of Eatough for further scheduling the events based on the bandwidth (*i.e. throughput rate*) availability. The modification would have been obvious to an ordinary skill in the art because it would provide an optimal method for processing each given task.

As per claim 30:

Colle teaches

associating objects based on one or more operational objects (see **FIGS. 3 and 5-8** – *Note: objects on the GUI are operational objects*).

As per claim 31

Colle teaches

wherein the operational objects correspond to at least a loop, a trigger, a conditional and hard start and/or stop times (see at least **FIGS. 3 and 5-8 –** “**Start Date, Start Time, End Date, and End Time**”).

<u>As per claim 39:</u>

Colle teaches

wherein information about event start and stop times and event duration is communicated with a particular object (see at least **FIGS. 3 and 5-8** – *Note: start date, start time, end date, and end time communicated with a particular task*).

<u>As per claim 44:</u>

Colle teaches

wherein the temporal constraints comprises one specific event must finished before another specific event starts (see at least Para [0059] “**each task may be identified with an additional scheduling constraint of either “as soon as possible” or “as late as possible**”).

<u>As per claim 45:</u>

Colle teaches

wherein the past and future space provide a context for navigation to a user during application development (see at least **FIGS. 3, 5-8, and 11**).

8. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Colle et al. (U.S. Publication No. 2004/0133889, hereinafter Colle), in view of Eatough (U.S. Publication. No. 2002/0087623), and in further view of Green et al. (U.S. Patent No. 4,646,231).

As per claim 11:

Neither Colle nor Eatough teaches

non-associated objects are executed randomly.

However, Green teaches

non-associated objects are executed randomly (see at least col. 1:60-67 "**In the present invention, a method of synchronizing the sequence by which a variety of unrelated activities are executed in a digital processor when those activities are randomly called by multiple caller includes the steps of providing a single processor queue for holding respective pointers to each different kind of activity that the processor performs...**").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify the teaching of Colle in combination with Eatough to incorporate the teaching of Green to allow objects to be executed randomly. The modification would have been obvious to one of ordinary skill in the art because it

would avoid the complication as the number of processors, tasks, and activities within each task increases.

9. Claims 25 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Colle et al. (U.S. Publication No. 2004/0133889, hereinafter Colle), in view of Eatough (U.S. Publication. No. 2002/0087623), and in further view of Zeidman (U.S. Patent No. 6,934,947).

As per claims 25 and 36:

Neither Colle nor Eatough teaches

the design component associates object via iterative loops.

However, Zeidman teaches

the design component associates object via iterative loops (see at least **FIGS. 1-5** – “**polling loop**”).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify the teaching of Colle in combination with Eatough to include the teaching of Zeidman to include loop task. The modification would have been obvious to one having an ordinary skill in the art because it would provide less memory space for task management.

10. Claims 40-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Colle et al. (U.S. Publication No. 2004/0133889, hereinafter Colle), in view of Eatough (U.S. Publication. No. 2002/0087623), and in further view of Zweben et al. (U.S. Patent No. 5,768,586).

As per claim 40:

Colle teaches

wherein the objects are bars (see at least **FIGS. 3, 7, and 11**).

Neither Colle nor Eatough teach

fuzzy edges on the bars indicate an unspecified time.

However, Zweben teaches

fuzzy edges on the bars indicate an unspecified time (See at least **FIG. 6**).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify the teaching of Colle in combination with Eatough to include the teaching of Zweben to include bar with fuzzy edge indicating unspecified time. The modification would have been obvious because it would be a user option for designing graphical user interface.

As per claim 41:

Zweben teaches

wherein the fuzzy edges on at beginning of the bar indicate an unspecified start time and the fuzzy logic on at end of the bar indicates an unspecified end time and/or duration (see at least **FIG. 6**).

As per claim 42:

Colle teaches

wherein hard bold edges on the bar specified specific start and/or stop time (see at least **FIGS. 3, 7, and 11**).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
6/1/2009

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191